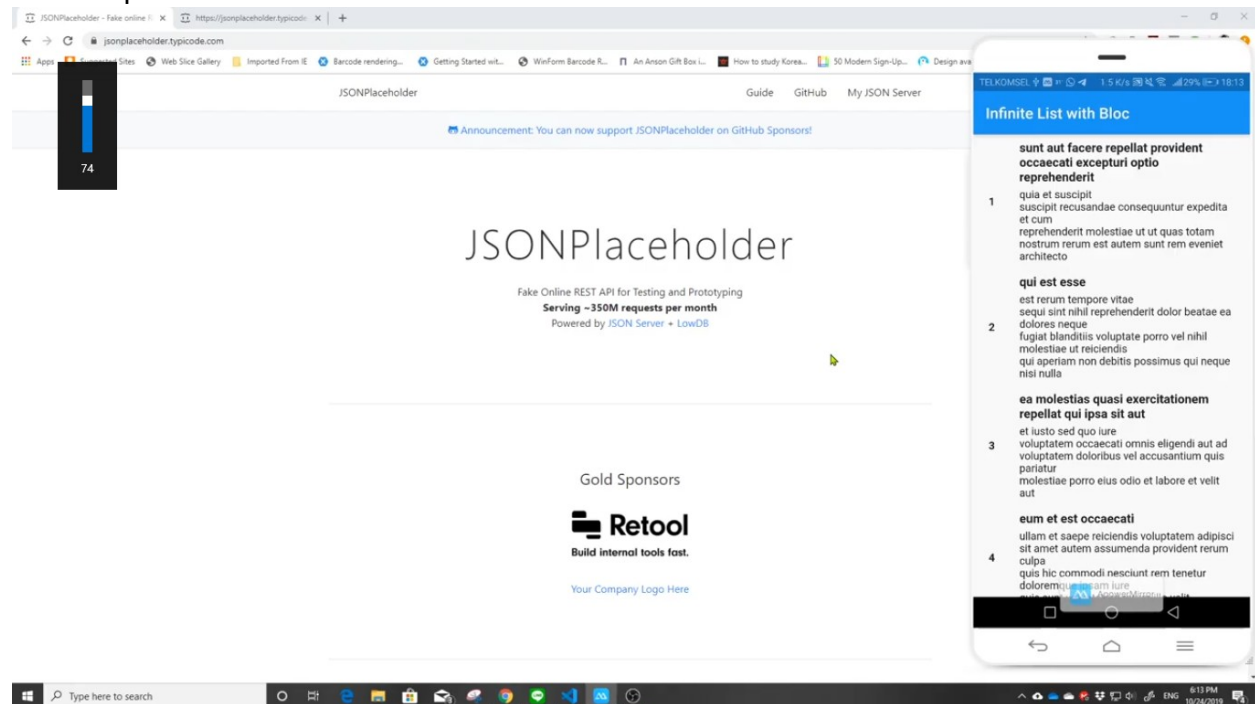
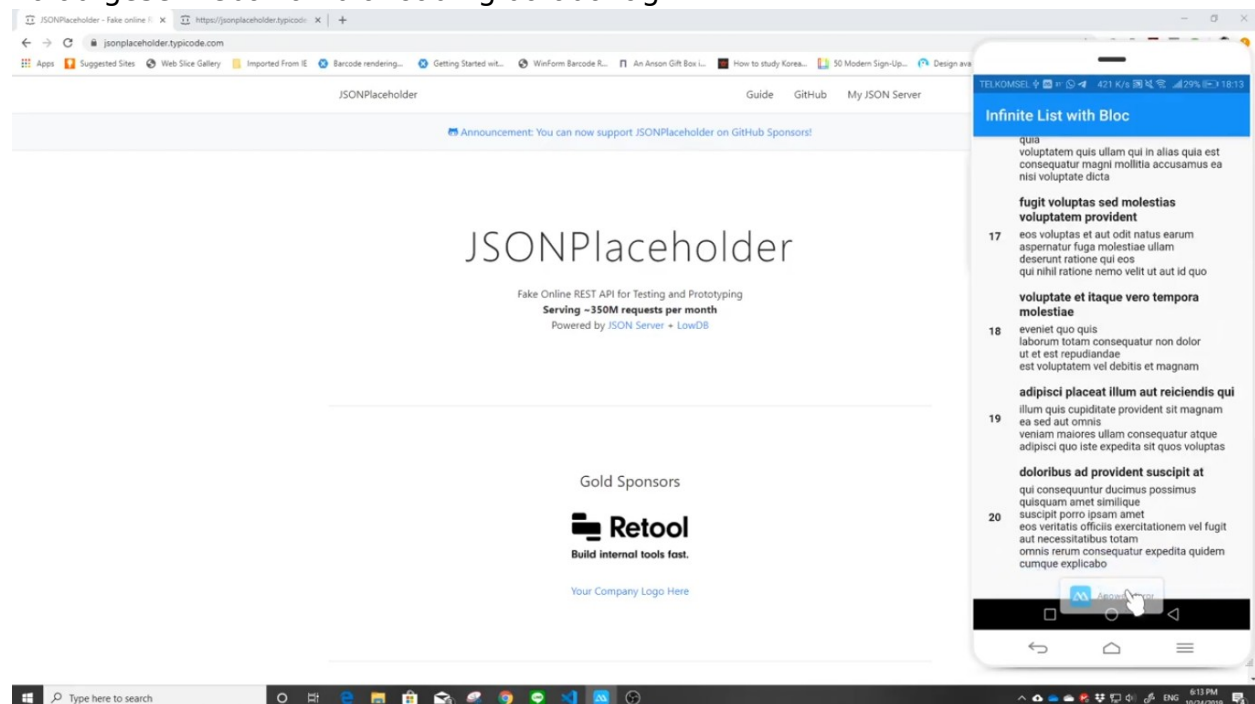


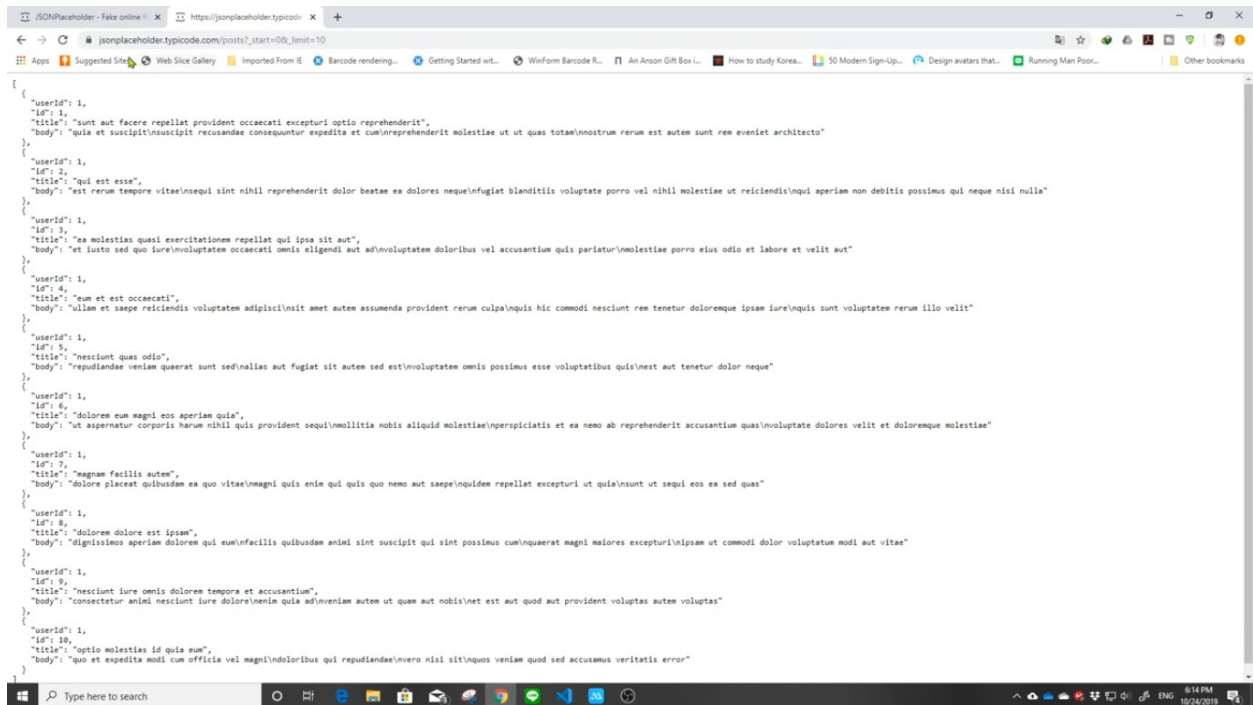
Menunjukkan bagaimana membuat infinite list dgn menggunakan state management bloc
Buat seperti ini



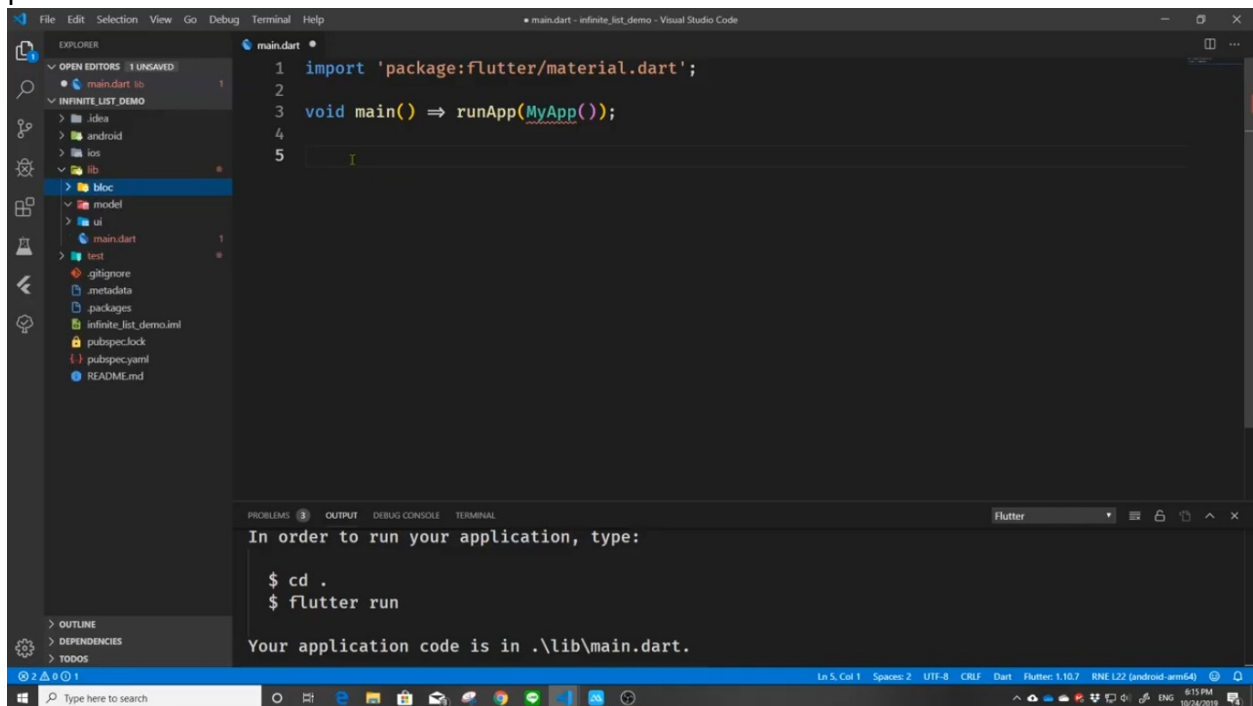
Kalau geser kebawah dia loading da ada lagi



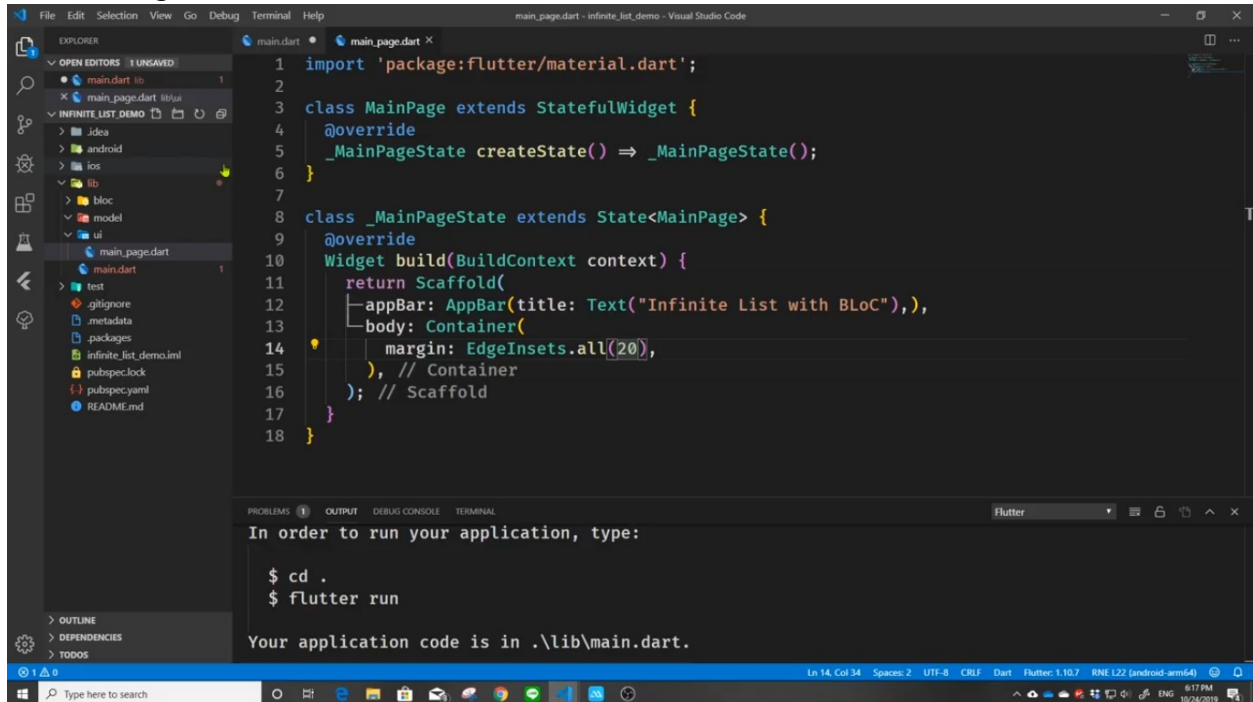
Untuk datanya diambil dari jsonplaceholder typin
Nanti bentuk url nya seperti ini



url nya
 post? Lalu satrt ada berapa ini 0 dan limit nya 10
 maka dapat 10 buah data data
 ada
 userid title body
 yg ipakai id title dan body
 langsung saja
 pertama buat folder model ada ui dan bloc



Lalu di main.dart nya
Buat stl
Home nya main page
Buat di ui mainpage.dart
Buat stf
Body nya pasang container dulu
Kasih margin 20



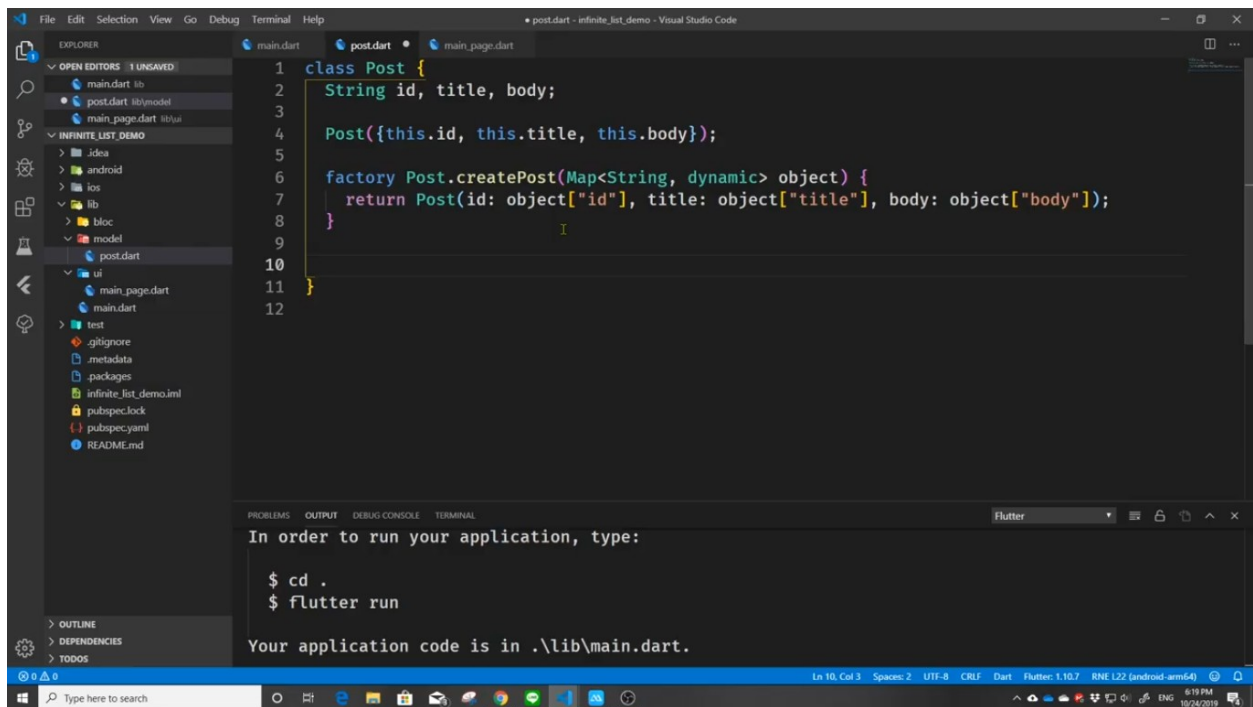
```
1 import 'package:flutter/material.dart';
2
3 class MainPage extends StatefulWidget {
4   @override
5   _MainPageState createState() => _MainPageState();
6 }
7
8 class _MainPageState extends State<MainPage> {
9   @override
10  Widget build(BuildContext context) {
11    return Scaffold(
12      appBar: AppBar(title: Text("Infinite List with BLoC")),
13      body: Container(
14        margin: EdgeInsets.all(20),
15      ), // Container
16    ); // Scaffold
17  }
18 }
```

In order to run your application, type:

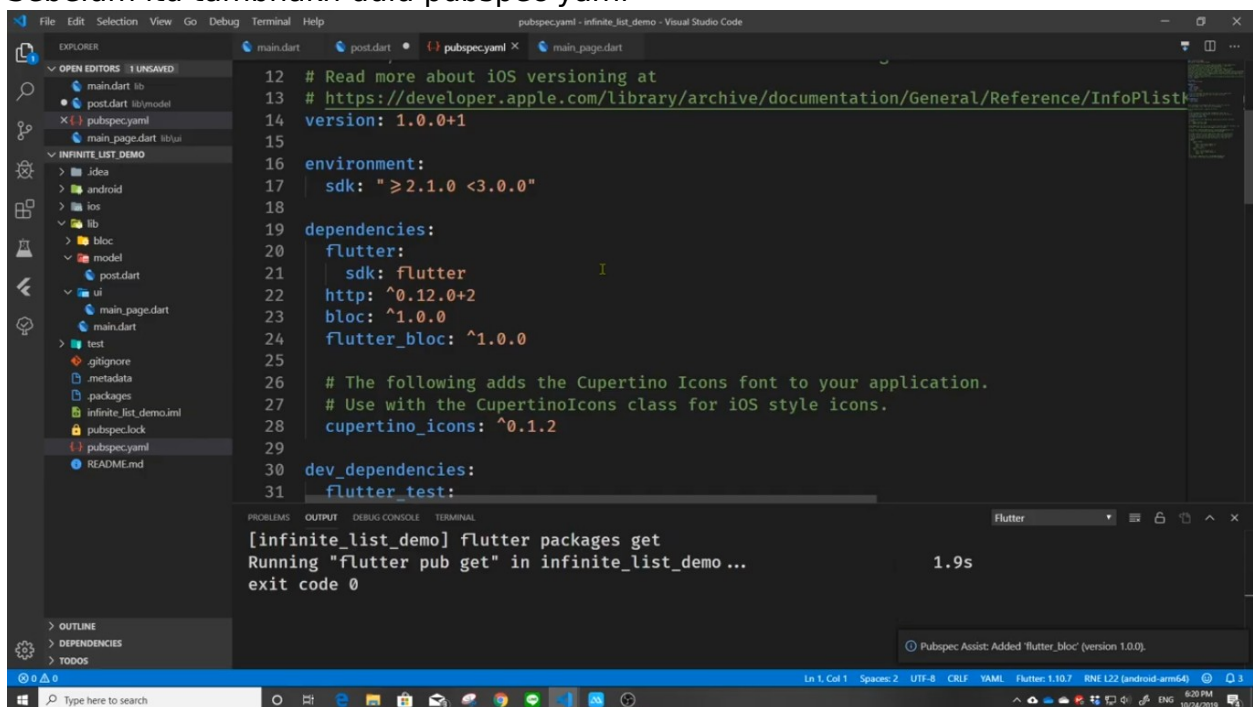
```
$ cd .
$ flutter run
```

Your application code is in .\lib\main.dart.

Main page import dahulu
Kalau sudah pertama kita buat dulu models nya
Buat post.dart
Buat class post nya dahulu
Punya field id title body lalu constructor nya
Lalu buat factory nya
Berikut nya buat koneksi api nya

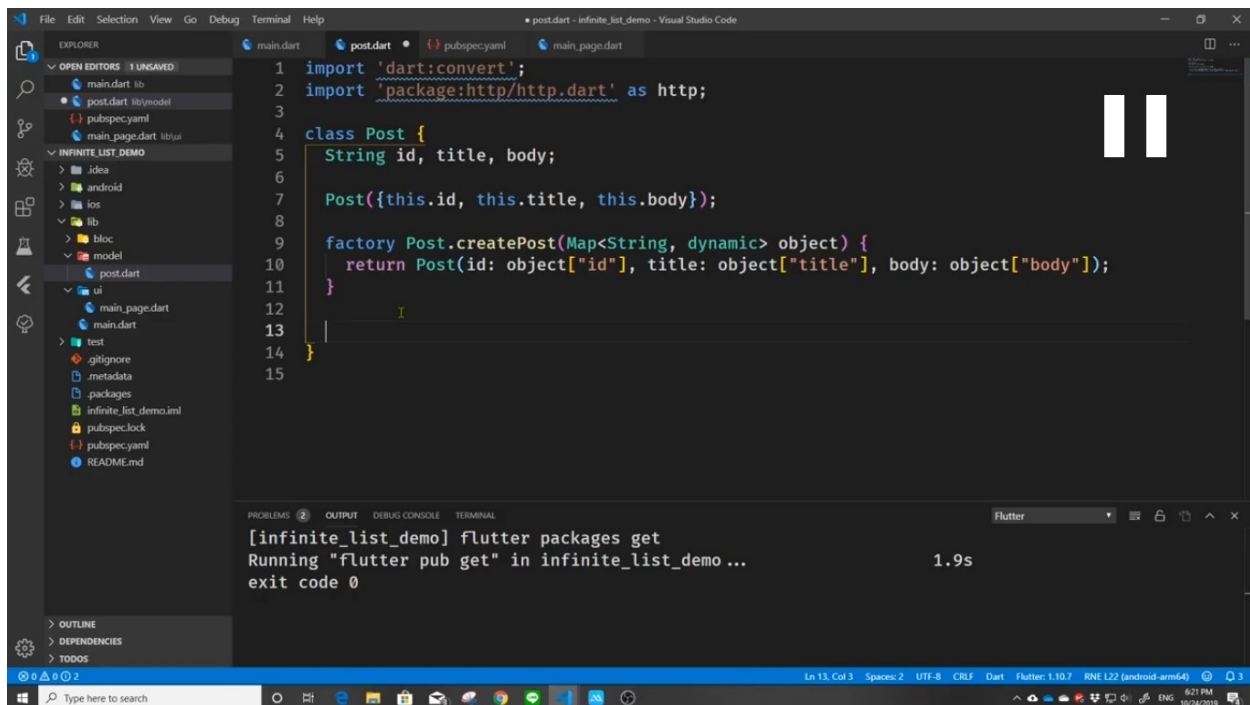


Sebelum itu tambhakan dulu pubspec.yaml



Pakai ctrl + p

Kalau sudah kembali ke post
Import dahulu packagenya

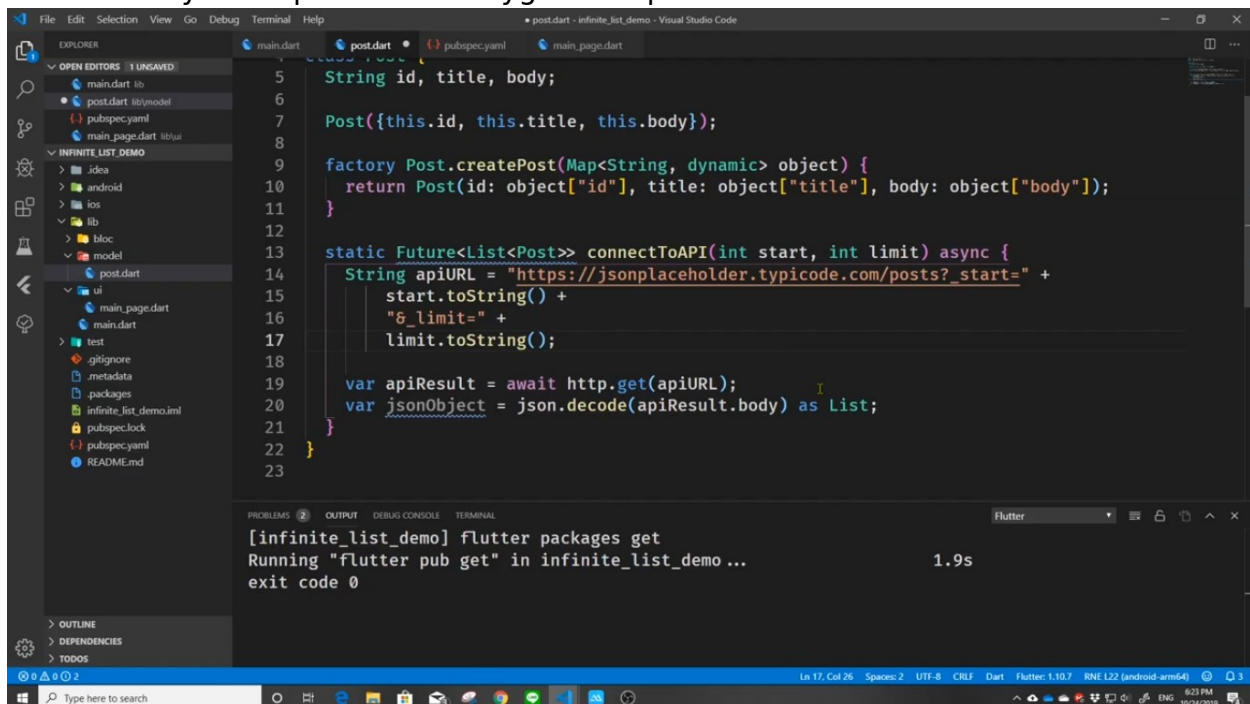


```
1 import 'dart:convert';
2 import 'package:http/http.dart' as http;
3
4 class Post {
5   String id, title, body;
6
7   Post({this.id, this.title, this.body});
8
9   factory Post.createPost(Map<String, dynamic> object) {
10     return Post(id: object["id"], title: object["title"], body: object["body"]);
11   }
12
13
14 }
15
```

Flutter

[infinite_list_demo] flutter packages get
Running "Flutter pub get" in infinite_list_demo ... 1.9s
exit code 0

Sekarang buat fungsi apinya
Dia kembalikan dia pakai future
Kita butuh start nya
Dan limit
Dan api url nya
Lalu panggil http.get dapat apiresult
Lalu ambil jsonObject dgn apiresult.body
Nanti hasilnya berupa list karena yg kita dapat dari list berita

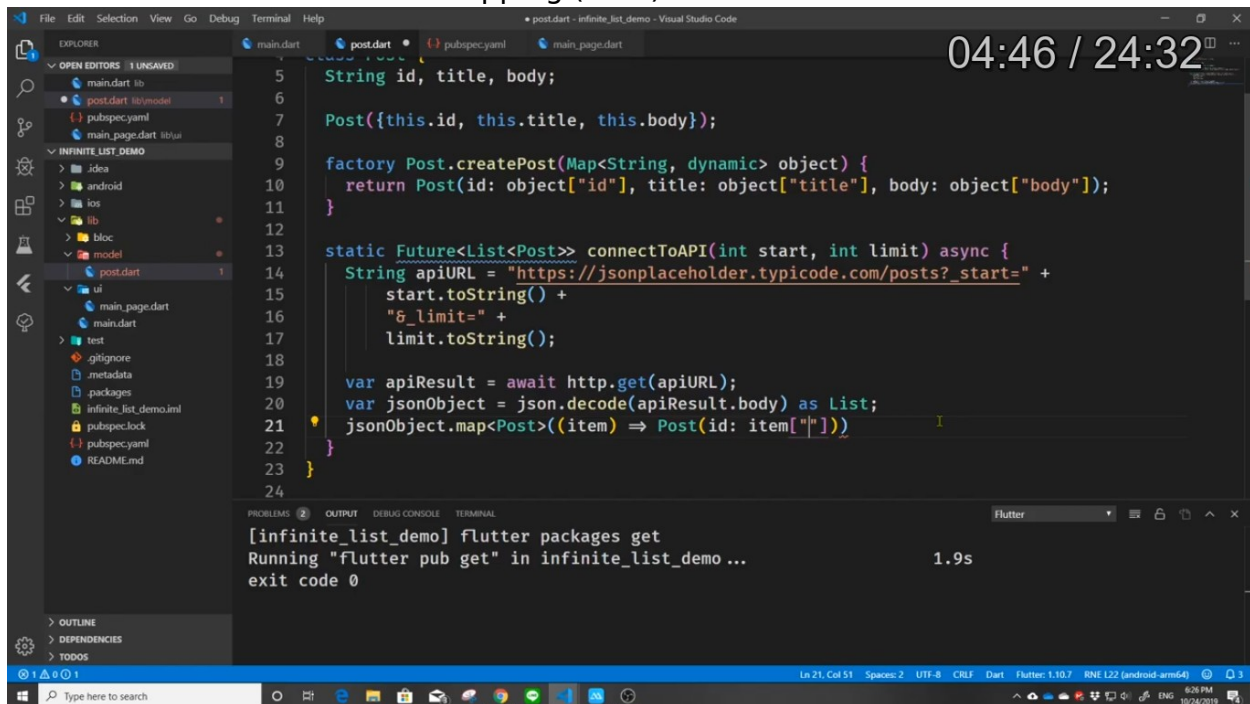


```
5   String id, title, body;
6
7   Post({this.id, this.title, this.body});
8
9   factory Post.createPost(Map<String, dynamic> object) {
10     return Post(id: object["id"], title: object["title"], body: object["body"]);
11   }
12
13   static Future<List<Post>> connectToAPI(int start, int limit) async {
14     String apiURL = "https://jsonplaceholder.typicode.com/posts?_start=" +
15       start.toString() +
16       "&_limit=" +
17       limit.toString();
18
19     var apiResult = await http.get(apiURL);
20     var jsonObject = json.decode(apiResult.body) as List;
21   }
22
23
```

Flutter

[infinite_list_demo] flutter packages get
Running "flutter pub get" in infinite_list_demo ... 1.9s
exit code 0

Berikutnya kita perlu mapping dari jsonobject yg merupakan list berita
List of post
Dia akan mereturn dari hasil mapping (item) disebut

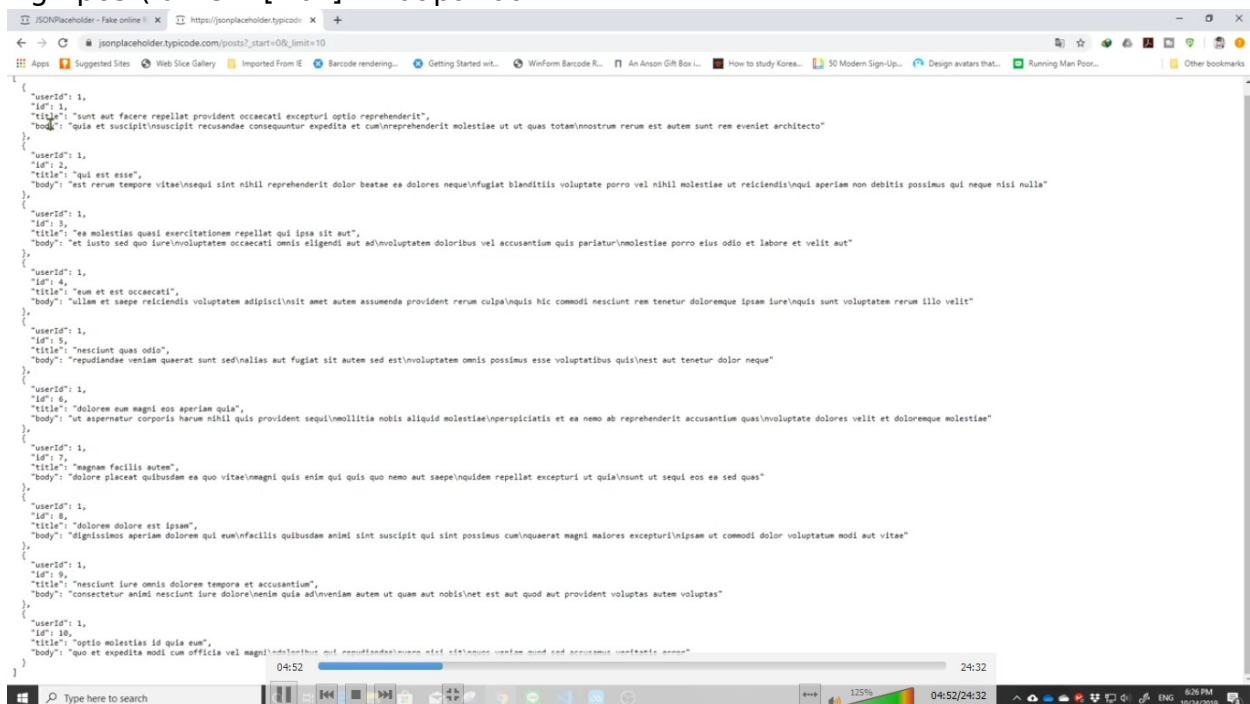


```
5 String id, title, body;
6
7 Post({this.id, this.title, this.body});
8
9 factory Post.createPost(Map<String, dynamic> object) {
10   return Post(id: object["id"], title: object["title"], body: object["body"]);
11 }
12
13 static Future<List<Post>> connectToAPI(int start, int limit) async {
14   String apiURL = "https://jsonplaceholder.typicode.com/posts?_start=" +
15     start.toString() +
16     "&_limit=" +
17     limit.toString();
18
19   var apiResult = await http.get(apiURL);
20   var jsonObject = json.decode(apiResult.body) as List;
21   jsonObject.map<Post>((item) => Post(id: item["id"]))
22 }
23
24 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL Flutter

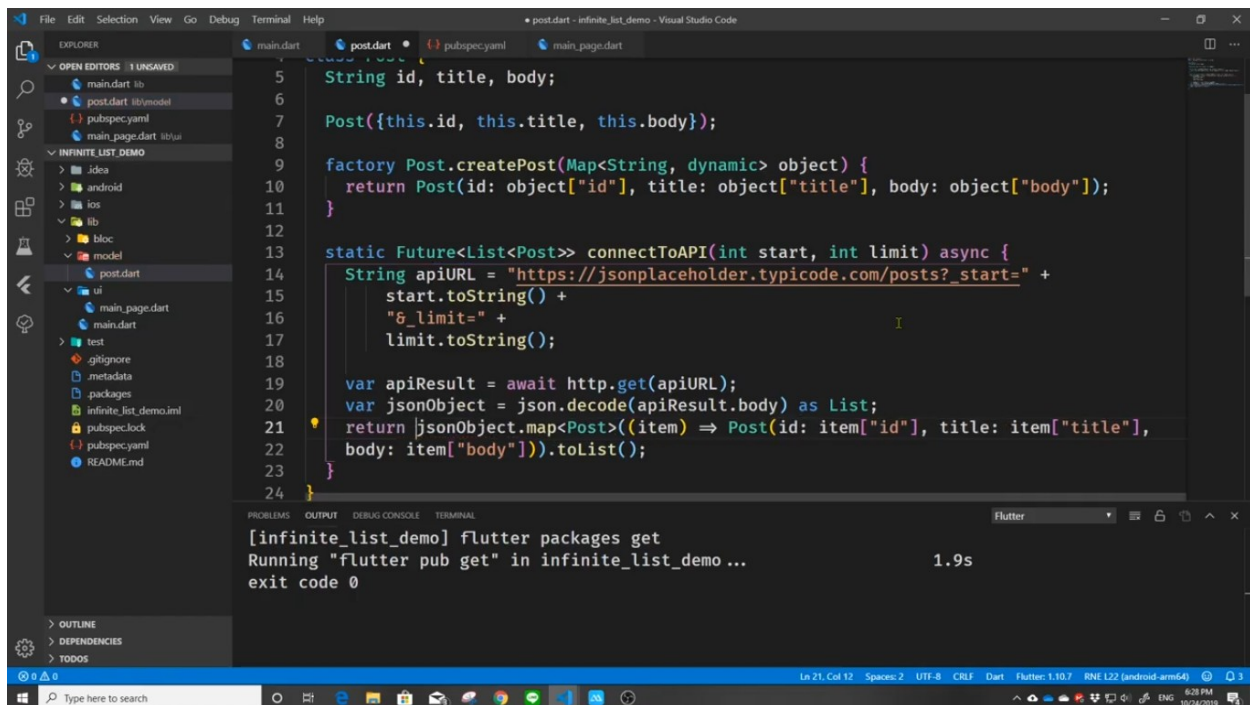
[infinite_list_demo] flutter packages get
Running "Flutter pub get" in infinite_list_demo ... 1.9s
exit code 0

Dgn post(id item:["id"]) ini dapat dari



```
{
  "userId": 1,
  "id": 1,
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
  "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
},
{
  "userId": 1,
  "id": 2,
  "title": "qui est esse",
  "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\ninfigiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\ninqui aperiam non debitis possimus qui neque nisi nulla"
},
{
  "userId": 1,
  "id": 3,
  "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
  "body": "et justo sed quo lore\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem doloribus vel accusantium quis pariatur\nmolestiae porro eius odio et labore et velit aut"
},
{
  "userId": 1,
  "id": 4,
  "title": "eum et est occaecati",
  "body": "ullam et saepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum culpa\nquis hic commodi nesciunt rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo velit"
},
{
  "userId": 1,
  "id": 5,
  "title": "nesciunt quas odio",
  "body": "repudiandae veniam quaerat sunt sed\nnihil aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nnec aut tenetur dolor neque"
},
{
  "userId": 1,
  "id": 6,
  "title": "dolorem eum magni eos aperiam quia",
  "body": "ut aspernatur corpora harum nihil quis provident sequi\nmollitia nobis aliquid molestiae\nperspiciatis et ea nemo ab reprehenderit accusantium quas\nvoluptate dolores velit et doloremque molestiae"
},
{
  "userId": 1,
  "id": 7,
  "title": "magnam facillit autem",
  "body": "dolore placeat quibusdam ea quo vitae\nmagni quis enim qui quis quo nemo aut saepe\ninqueam repellat excepturi ut quia\nsunt ut sequi eos ea sed quas"
},
{
  "userId": 1,
  "id": 8,
  "title": "dolorem dolore est ipsam",
  "body": "dignissimos aperiam dolorem qui eum\nfacilis quibusdam animi sint suscipit qui sint possimus cum\nquaerat magni maiores excepturi\nipsam ut commodi dolor voluptatum modi aut vitae"
},
{
  "userId": 1,
  "id": 9,
  "title": "nesciunt iure omnis dolorem tempora et accusantium",
  "body": "consectetur animi nesciunt iure dolore\nnam quia ad\nveniam autem ut quam aut nobis\net est aut quod aut provident voluptas autem voluptas"
},
{
  "userId": 1,
  "id": 10,
  "title": "optio molestias id quia eum",
  "body": "quo et expedita modi cum officia vel magni\ndoloribus ut reiciendis\nvoluptatem aut dignissimos debitis cumque nihil aletum enim\nquia culpa et accusamus molestias nonnulla"
}
}
```

Sini ada id title body ini yg kita dapat
Dapat idn ya lalu masukan ke post ini
Lalu seteah dimaping kita berupoa menjadi list
Hasil ini yg akan kita return



```
File Edit Selection View Go Debug Terminal Help
post.dart - infinite_list_demo - Visual Studio Code

EXPLORER
main.dart lib
post.dart lib/model
pubspec.yaml
main_page.dart lib/ui
INFINITE_LIST_DEMO
idea
android
ios
lib
bloc
model
post.dart
ui
main_page.dart
main.dart
test
gignore
metadata
packages
infinite_list_demo.iml
pubspec.lock
pubspec.yaml
README.md

main.dart
5 String id, title, body;
6
7 Post({this.id, this.title, this.body});
8
9
10 factory Post.createPost(Map<String, dynamic> object) {
11   return Post(id: object["id"], title: object["title"], body: object["body"]);
12 }
13
14 static Future<List<Post>> connectToAPI(int start, int limit) async {
15   String apiURL = "https://jsonplaceholder.typicode.com/posts?_start=" +
16     start.toString() +
17     "&_limit=" +
18     limit.toString();
19
20   var apiResult = await http.get(apiURL);
21   var jsonObject = json.decode(apiResult.body) as List;
22   return jsonObject.map<Post>((item) => Post(id: item["id"], title: item["title"],
23     body: item["body"])).toList();
24 }

TERMINAL
[flutter] flutter packages get
Running "Flutter pub get" in infinite_list_demo...
exit code 0
1.9s
```

sekarang pindah ke bloc

Kita buat post bloc dart

Disini kita buat postEvent kita kosongkan {}

Lalu buat class posState pakai abstract class

Dan buat lagi turunan dari PostState

Yg pertama

PostInitiallized ini kita pakai data yg pertma akali

Dan postLoaded ini ketika kita sudah load datnya

Didalm postLoaded ini kita simapn List<Post> post;

Lalu buat buat boolean hasReachedmax ini digunakan untuk apakah kita load semua data

Yg ada di ui kalau gk ada datanya lagi maka ini menjadi true

Berikutnya buat constructor nya

Lalu untuk mempermudah pekerjaan kita nanti maka buat buah method

postLoaded copyWith ({dia minta Lis<Post> post, bool }){

yg dia return postLoaded yg baru post itu kita isi post kalau post ini tidak nul

post ini paramter kalau null maka kembalikan post dari field

lalu hasreack : has juga sama

The screenshot shows the Visual Studio Code editor with the file explorer on the left. The 'post_bloc.dart' file is open, showing the following code:

```
1 import 'package:infinite_list_demo/model/post.dart';
2
3 class PostEvent {}
4
5 abstract class PostState {}
6
7 class PostUninitialized extends PostState {}
8
9 class PostLoaded extends PostState {
10   List<Post> posts;
11   bool hasReachedMax;
12
13   PostLoaded({this.posts, this.hasReachedMax});
14
15   PostLoaded copyWith({List<Post> posts, bool hasReachedMax}) {
16     return PostLoaded(
17       posts: posts ?? this.posts,
18       hasReachedMax: hasReachedMax ?? this.hasReachedMax);
19   }
20 }
```

The terminal at the bottom shows the output of the command `[infinite_list_demo] flutter packages get`, indicating that the packages were successfully updated.

Ini baru buat postEvent dan PostState
Sekarang buat post Bloc nya

The screenshot shows the Visual Studio Code editor with the file explorer on the left. The 'post_bloc.dart' file is open, showing the following code:

```
10 List<Post> posts;
11 bool hasReachedMax;
12
13 PostLoaded({this.posts, this.hasReachedMax});
14
15 PostLoaded copyWith({List<Post> posts, bool hasReachedMax}) {
16   return PostLoaded(
17     posts: posts ?? this.posts,
18     hasReachedMax: hasReachedMax ?? this.hasReachedMax);
19 }
20
21
22 class PostBloc extends Bloc
23
```

A context menu is visible over the `PostBloc` class definition, showing options to 'Import library', 'Create class', or 'Create mixin'.

The terminal at the bottom shows the output of the command `[infinite_list_demo] flutter packages get`, indicating that the packages were successfully updated.

Import

Lalu masukan itu adalah <postEvent, kemablain adalah postState>

Initial state berarti PostUnitilezed

Lalu di stream ganti cek dulu

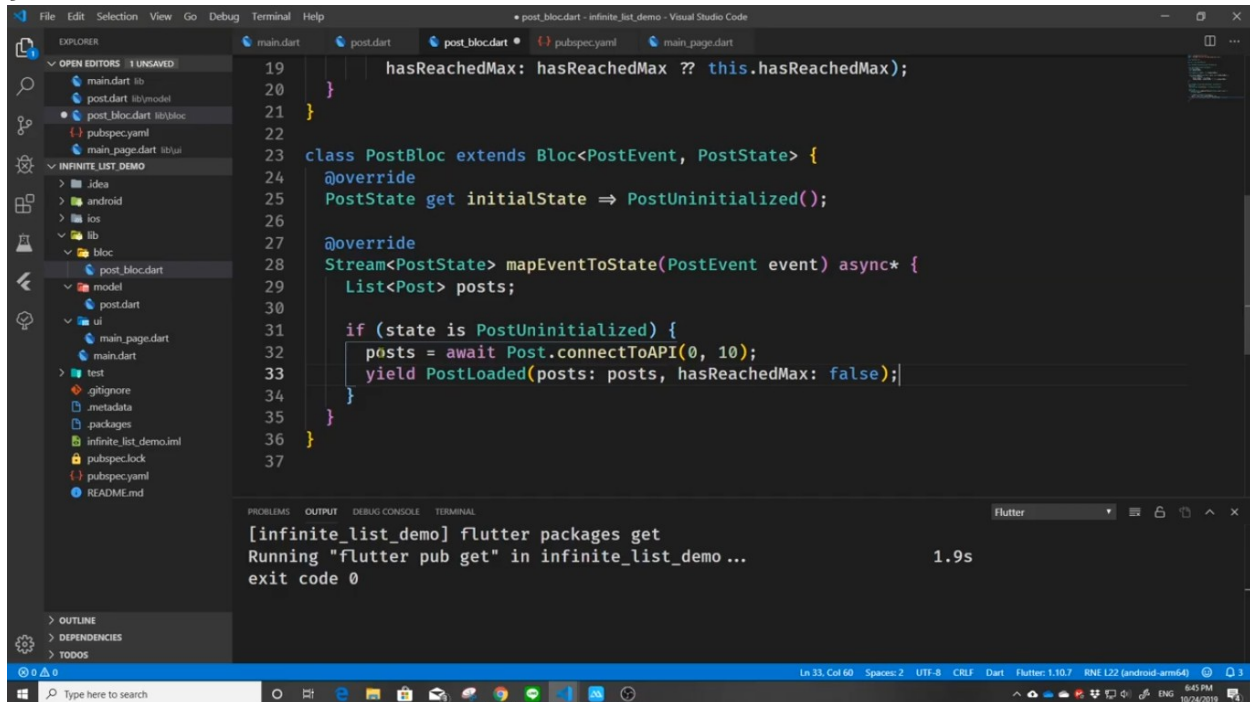
Buat List<Post> Post

If (jika state is PostUnitiled berarti belum pernah load data){


```

Posts= await post.connectToApi(satrt 0 , limit 10) ambil per 10
Kalau sudah kita yield PostLoaded(Posts= posts, hasReach itu false);
}

```



Ini perlu diketahui

Bloc versi 1

Sekarang pakai state is Post

Sebelumnya bukan State

Lalu else

Kalau bukan postUnitial berarti pernah load

Maka buat dahulu

postLoaded postloaded = state as Postloaded ini post saat ini

posts nya itu = ambil nya wait post.connect(mulai dari index terakhir

postLoaded.posts.length,

kalau sduah ambil 10 makaa index 0 sampai 9 berarti index berikutnya diambil 10)

sekarang kembalikan itu adalah yield kita cek dahlu

jika hasil dari api listkosong maka sudah habis sduah mentok

maka kembalikan data sebelumnya

cuman update post.coy(hasreachmax menjadi true) : kalau tidak kosong maka

kembalikan postloaded yg baru dari postloaded sekarang yg ada + dgn hasil dari

api nya lalu haslreach nya false

```
24 @override
25 PostState get initialState => PostUninitialized();
26
27 @override
28 Stream<PostState> mapEventToState(PostEvent event) async* {
29   List<Post> posts;
30
31   if (state is PostUninitialized) {
32     posts = await Post.connectToAPI(0, 10);
33     yield PostLoaded(posts: posts, hasReachedMax: false);
34   } else {
35     PostLoaded postLoaded = state as PostLoaded;
36     posts = await Post.connectToAPI(postLoaded.posts.length, 10);
37
38     yield (posts.isEmpty)
39       ? postLoaded.copyWith(hasReachedMax: true)
40       : PostLoaded(posts: postLoaded.posts + posts, hasReachedMax: false);
41   }
42 }
43 }
```

[infinite_list_demo] flutter packages get
Running "flutter pub get" in infinite_list_demo ... 1.9s
exit code 0

Jadi bloc seperti ini

Kalau sudah kembali ke main.dart

Di mainPage kibungkus dgn blocProvider

Import bloc tipenya <PostBloc>

Builder context => kembalikan postbloc

Dibloc versi 1.0 sibloc tidak pakai lagi dispanc skrang pakai .add

Tulis seperti ini supaya otomatis dipanggil data pertama kali ketika dibuat
postBlocini ambil data pertama

```
1 import 'package:flutter/material.dart';
2 import 'package:flutter_bloc/flutter_bloc.dart';
3 import 'package:infinite_list_demo/bloc/post_bloc.dart';
4 import 'package:infinite_list_demo/ui/main_page.dart';
5
6 void main() => runApp(MyApp());
7
8 class MyApp extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      debugShowCheckedModeBanner: false,
13      home: BlocProvider<PostBloc>(
14        builder: (context) => PostBloc()..add(PostEvent()),
15        child: MainPage(), // BlocProvider
16      ); // MaterialApp
17  }
18 }
```

[infinite_list_demo] flutter packages get
Running "flutter pub get" in infinite_list_demo ... 1.9s
exit code 0

Kita ke main page child nya kita buat
 blocbuilder<postbloc, postState>{ import
 builder : (context, state){
 disini ada 2 kemungkinan
 sistate masih initial atau postloaded
 kita if dulu
 jika satet is PostUniled)
 maka return seperti loading kalau belum diloat
 sekarang kalau sudah load
 maka else
 biar mudah
 postlaoded postlaoded = state as postlaoded; biar mudah
 lalu return adalah listView builder
 item count nya diambil poslaoded.post.lengt

```

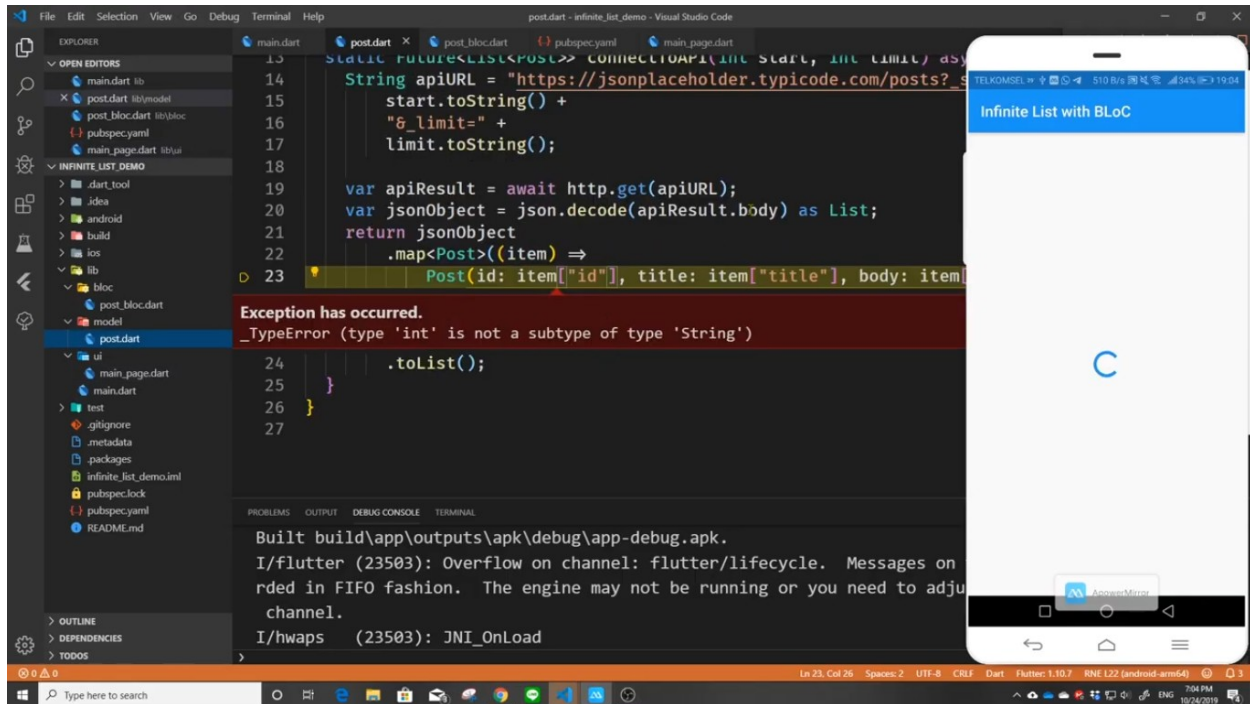
10 class _MainPageState extends State<MainPage> {
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       appBar: AppBar(title: Text("Infinite List with BloC")),
15       body: Container(
16         margin: EdgeInsets.all(20),
17         child: BlocBuilder<PostBloc, PostState>(
18           builder: (context, state) {
19             if (state is PostUninitialized)
20               return Center(child: SizedBox(width: 30, height: 30, child: CircularProgressIndicator));
21             else
22               {
23                 PostLoaded postLoaded = state as PostLoaded;
24                 return ListView.builder(
25                   itemCount: postLoaded.posts.length,
26                 ) // ListView.builder
27               }
28             },
29           ), // BlocBuilder

```

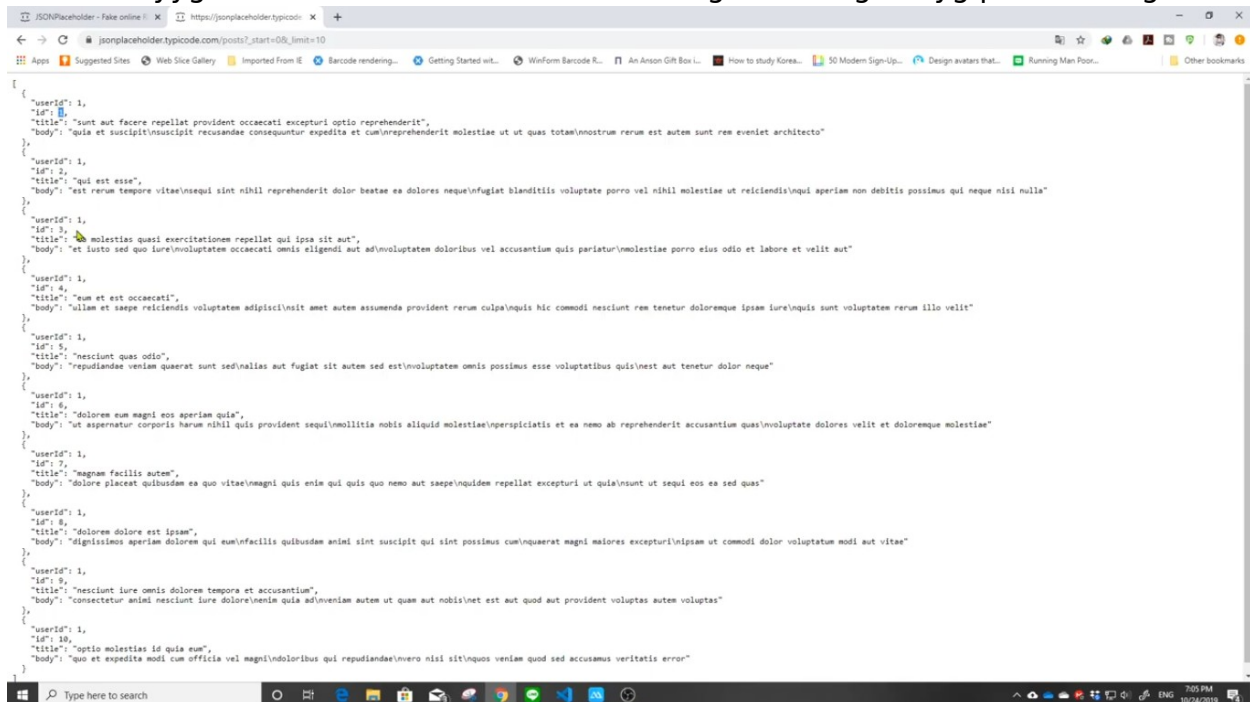
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

[infinite_list_demo] flutter packages get
 Running "flutter pub get" in infinite_list_demo ... 1.9s
 exit code 0

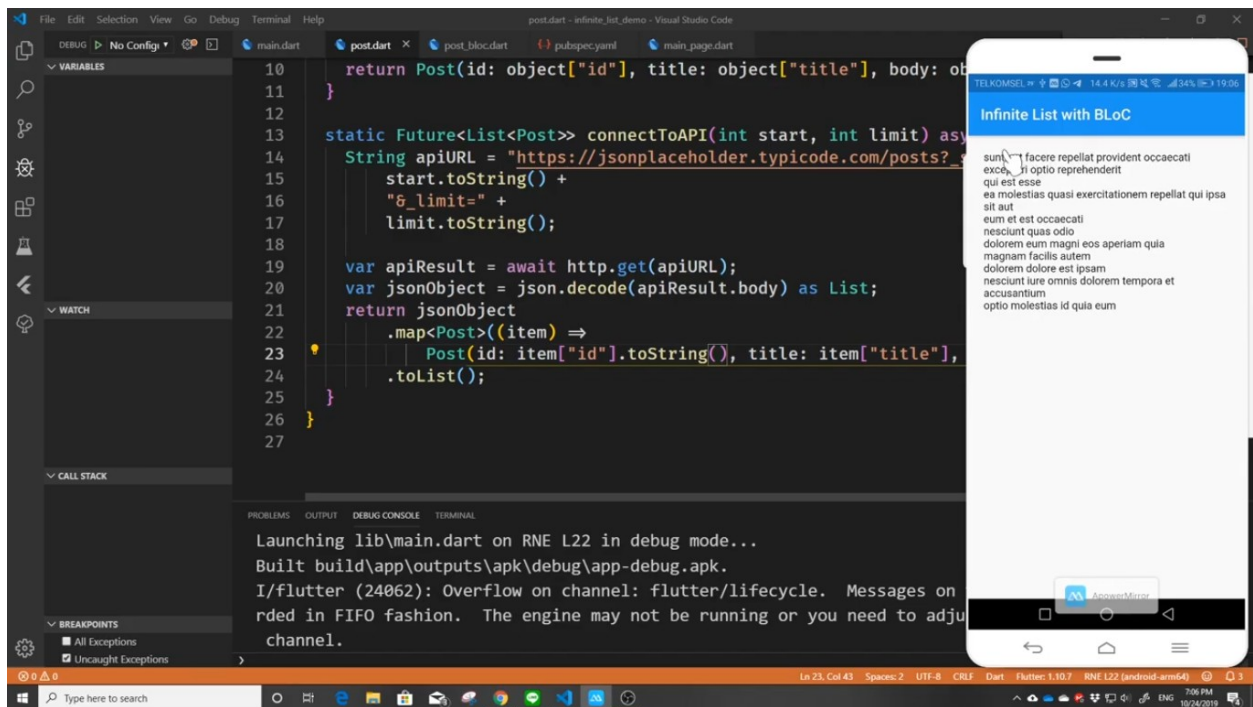
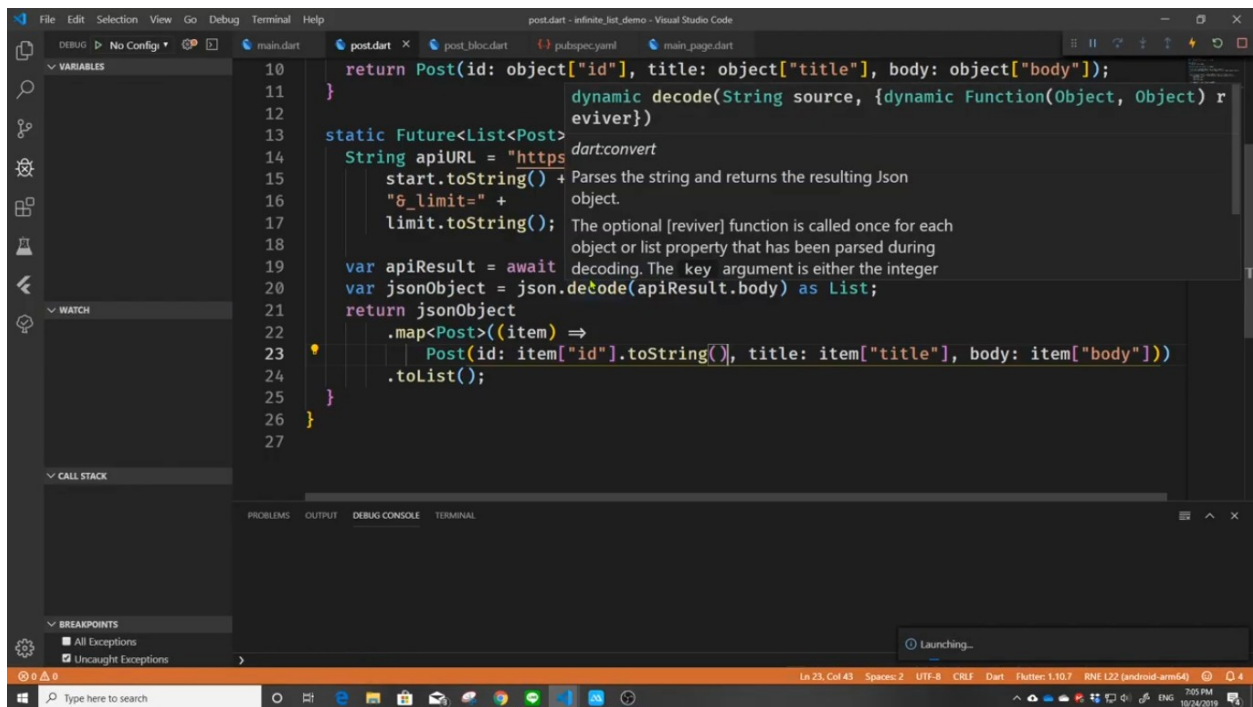
baru disini itemBuilder itu apa
 <context, index> => untuk saat ini judul nya dahulu
 Text postloaded.post[index].tile
 Coba jalankan



Disini ada yg salah kembalian disni adalah integer sedangkan yg perlu string



Kasih toString



Maka sudah
 Buat lebih bagus tampilan nya kembali lagi mainPage
 Hapus saja title nya kita ganti


```

21   if (state is PostUninitialized)
22     return Center(
23       child: SizedBox(
24         width: 30,
25         height: 30,
26         child: CircularProgressIndicator(),
27       ); // SizedBox // Center
28   else {
29     PostLoaded postLoaded = state as PostLoaded;
30     return ListView.builder(
31       itemCount: postLoaded.posts.length,
32       itemBuilder: (context, index) =>
33         Text(postLoaded.posts[index].title),
34     ); // ListView.builder
35   }
36 }, // BlocBuilder
37 ), // Container
38 ); // Scaffold
39 }
40 }

```

Launching lib\main.dart on RNE L22 in debug mode...
 Built build/app/outputs/apk/debug/app-debug.apk.
 I/flutter (24062): Overflow on channel: flutter/lifecycle. Messages on this channel are being discarded in FIFO fashion. The engine may not be running or you need to adjust the buffer size if of the channel.

Dgn PostItem(PostLoaded.Posts[Index]) buat dulu post item dart
 Buat stl PostItem
 Cobakan dahulu kembalikan nya text nya
 Buat field dan constroktor nya

```

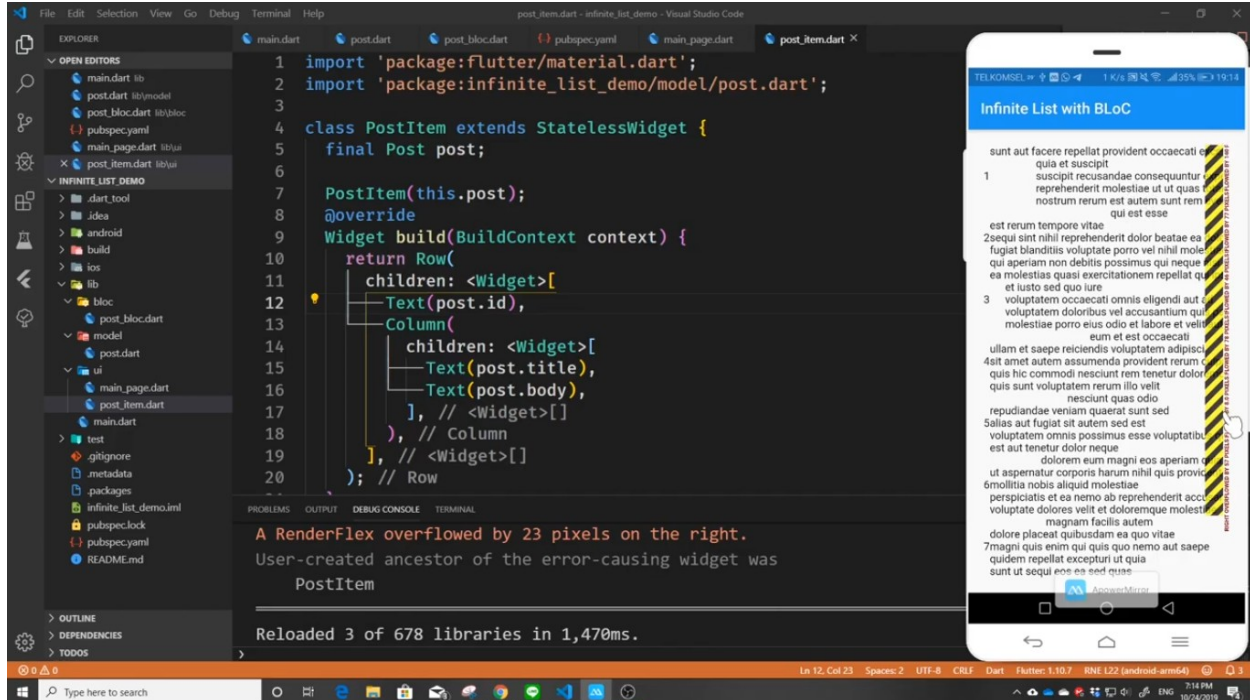
1 import 'package:flutter/material.dart';
2 import 'package:infinite_list_demo/model/post.dart';
3
4 class PostItem extends StatelessWidget {
5   final Post post;
6
7   PostItem(this.post);
8   @override
9   Widget build(BuildContext context) {
10    return Text(post.title);
11  }
12 }

```

Built build/app/outputs/apk/debug/app-debug.apk.
 I/flutter (24062): Overflow on channel: flutter/lifecycle. Messages on this channel are being discarded in FIFO fashion. The engine may not be running or you need to adjust the buffer size if of the channel.
 Reloaded 0 of 677 libraries in 487ms.

Lalu impor postItem nya
 Text nya kita bungkus row
 Karena nanti ada nomor nya disamping
 Lalu sebelah ada text title nya bawah nya body

Maka bungkus lagi column
Pasang title,dia atas columnb pasang id



Disini masih ada kelebihan maka kita rapikan

Row nya kita bungkus dgn container kasih margin top 8 bottom 8

MainAxis kasih bettween

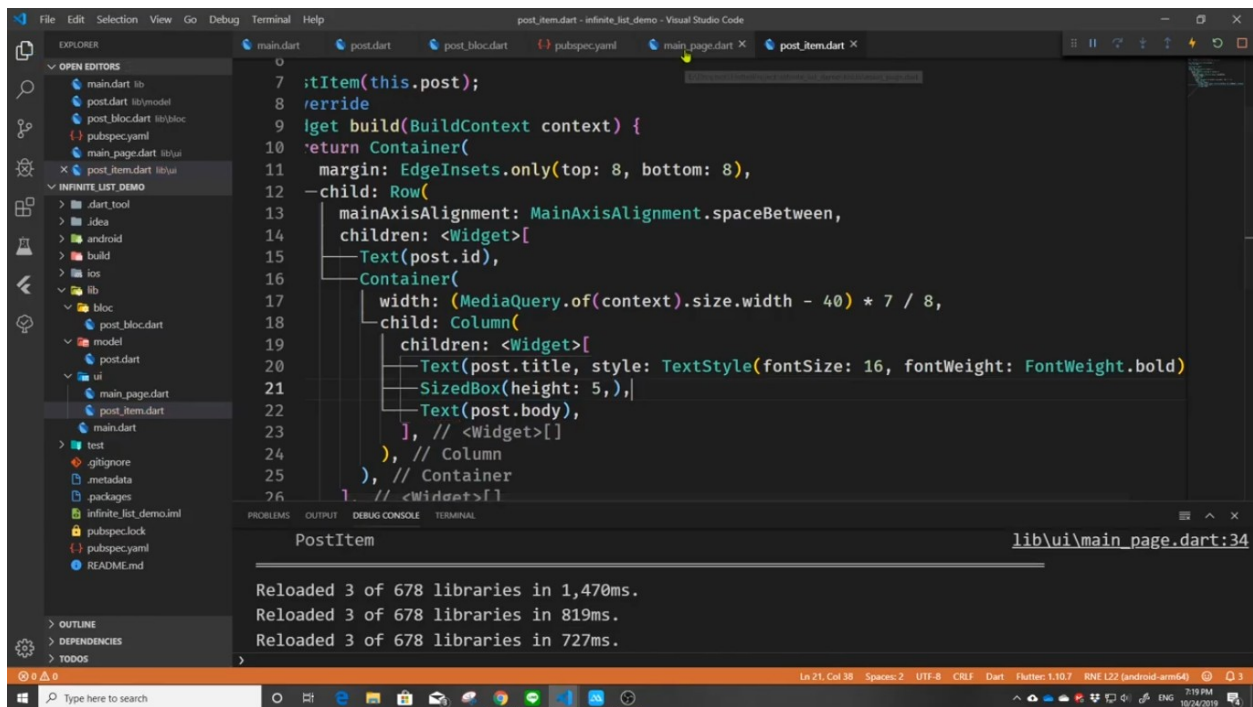
Lalu column nya bungkus lagi dgn container kasih

Widthn'ta media query.size.widhy kurang 40 karena ada margin kiri kanan 20 20

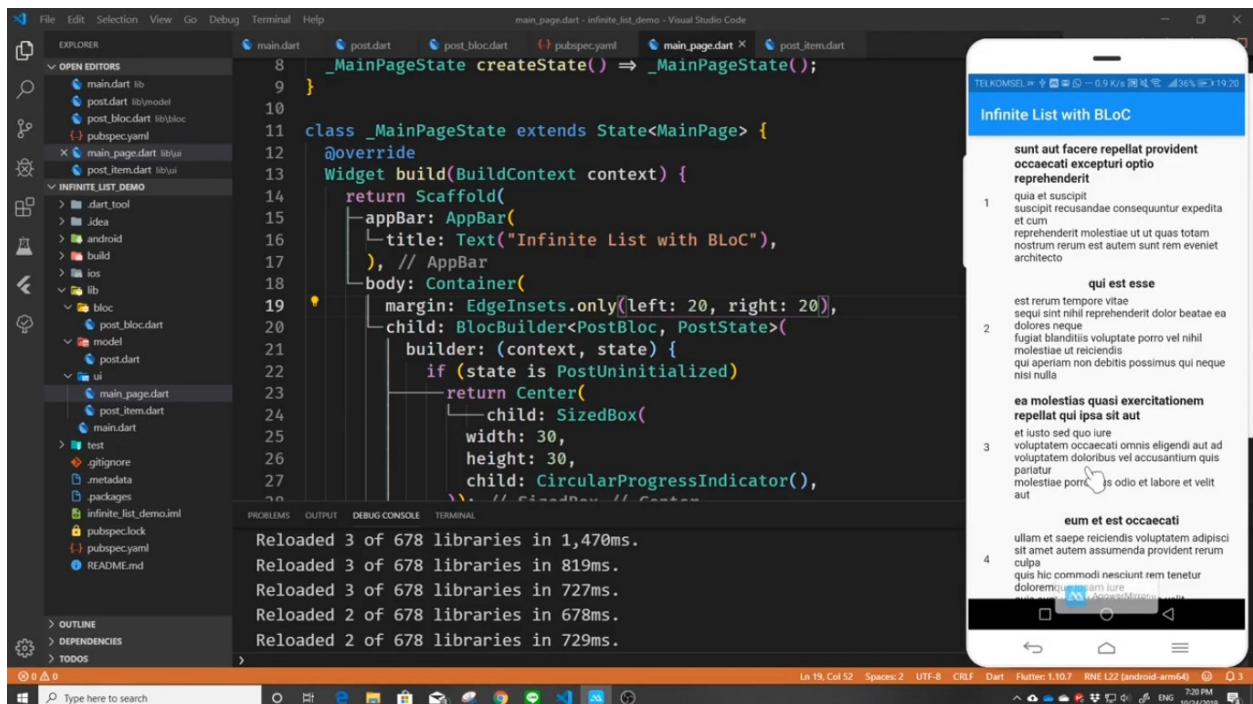
Lalu kali 7/8

Lalu kasih textStyleya

Dibawah text pasang sizedbox

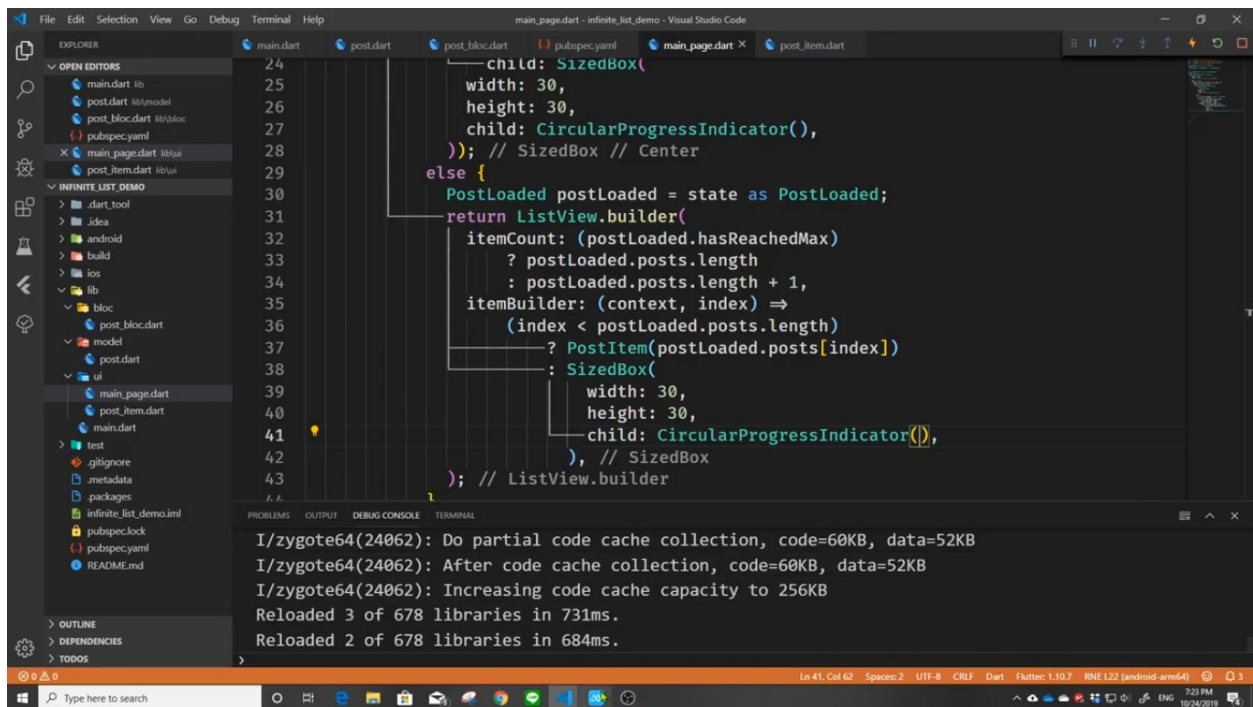


Lalu ke mainPage
 Disni jgn all tapi only left 20 right 20

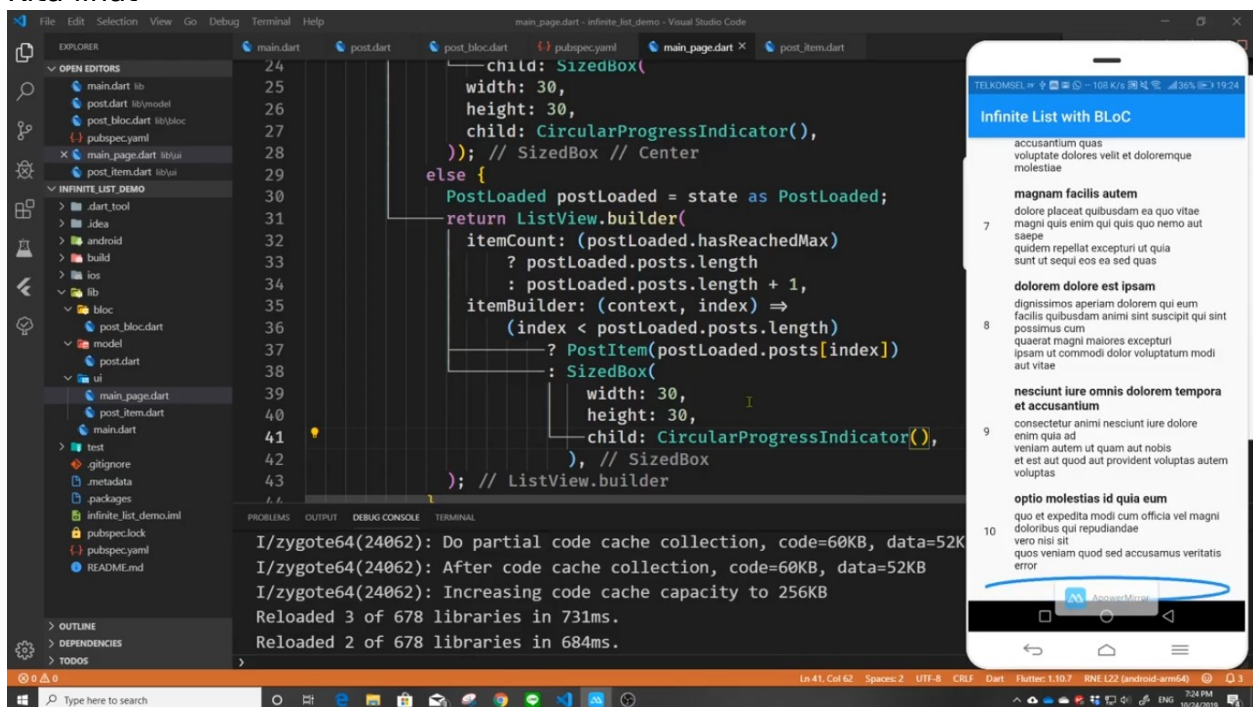


Lihat maka hasil sudah bagus
 Tingal geser judul nya jadi rata kiri



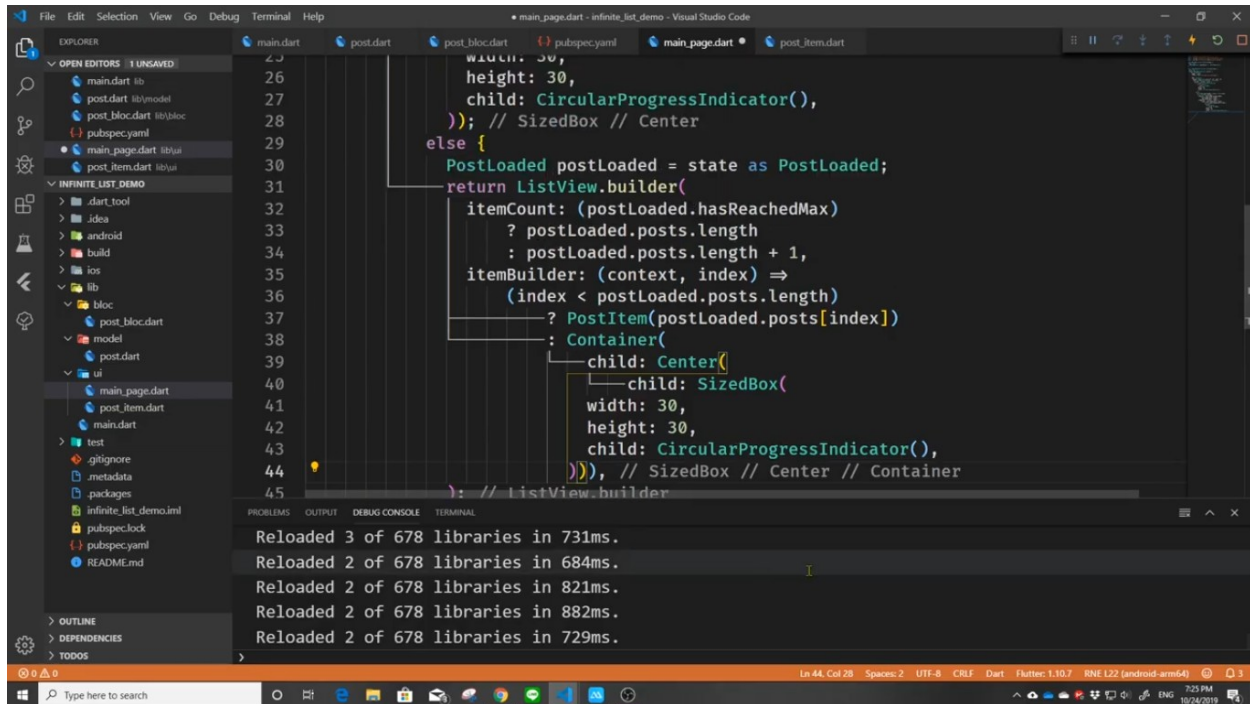


Kita lihat

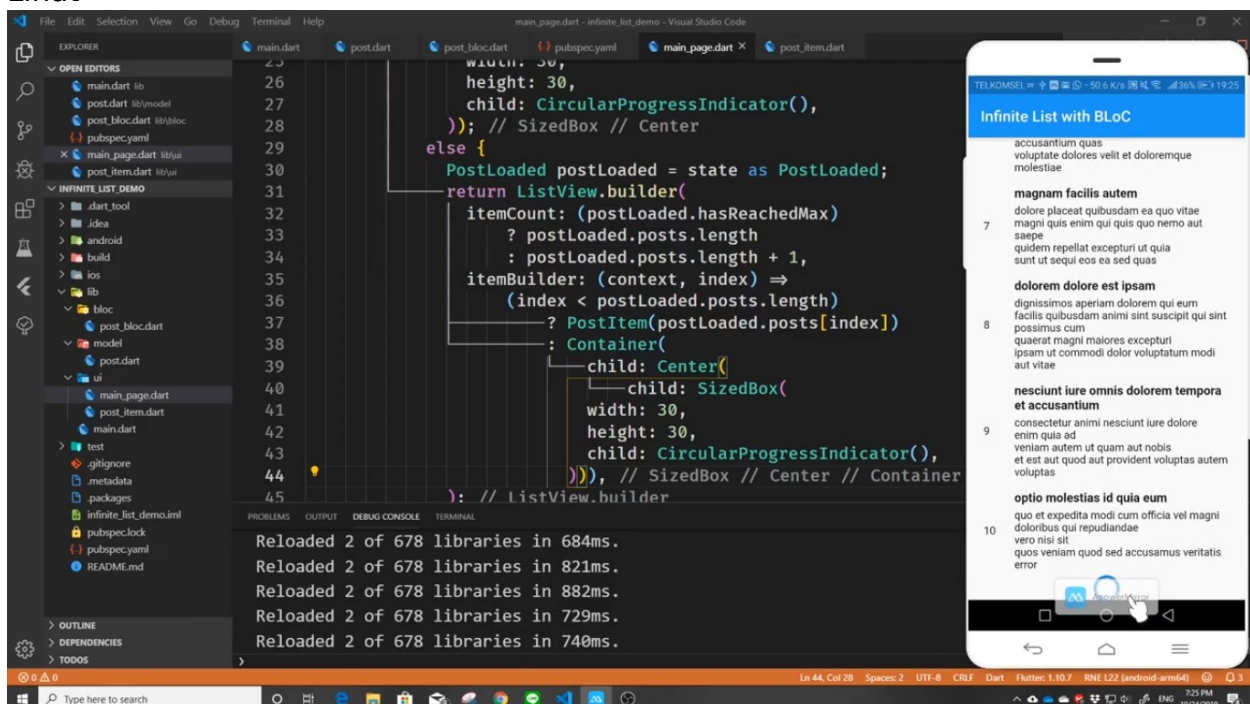


Ini masih panjang kita bungkus lagi sized box dgn container

Center



Lihat

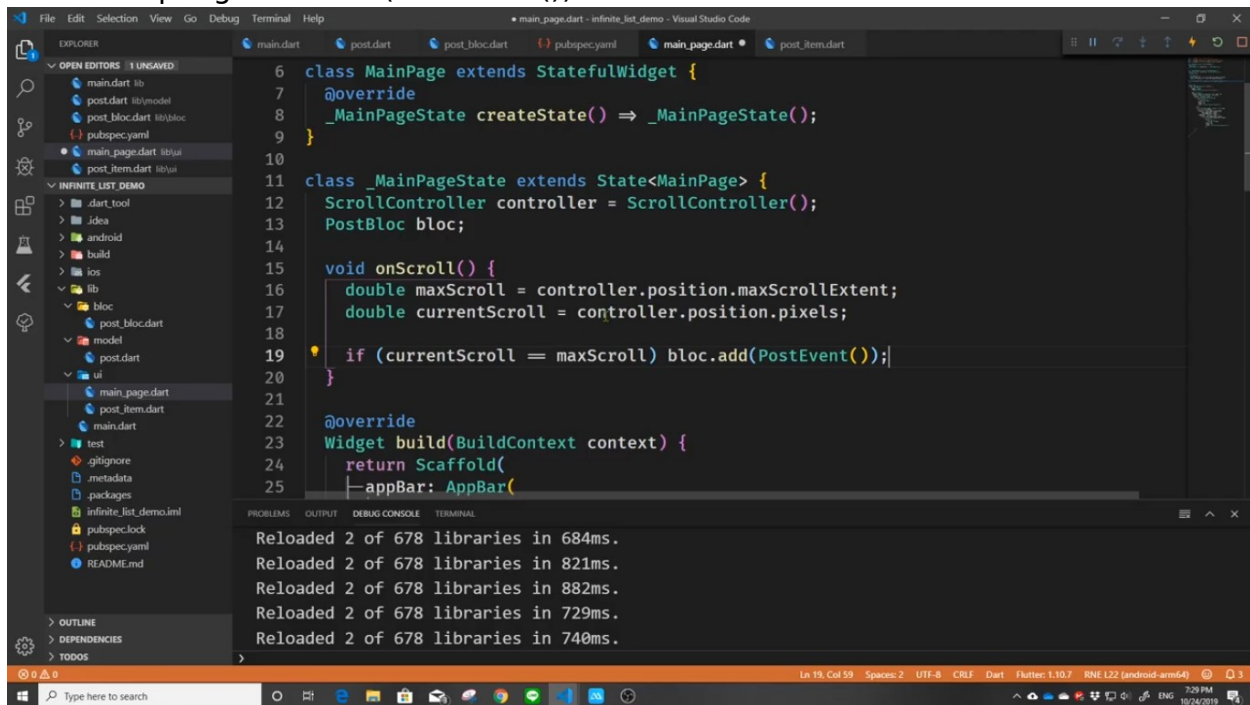


Tinggal buat scrollController controller = ScrollController;
Lalu PostBloc bloc;
Disini buat method
Void onScroll(){
Double maxScroll = controller.position.maxScrollExtent;

Lalu dibawahnya double currentScroll = controller.position.pixels

If (currentScroll berarti posisi scroll == ternyata sudah sama dgn samapai paling bawah maxScroll)

Maka kita panggil bloc.add(PostEvent())



```
6 class MainPage extends StatefulWidget {
7   @override
8   _MainPageState createState() => _MainPageState();
9 }
10
11 class _MainPageState extends State<MainPage> {
12   ScrollController controller = ScrollController();
13   PostBloc bloc;
14
15   void onScroll() {
16     double maxScroll = controller.position.maxScrollExtent;
17     double currentScroll = controller.position.pixels;
18
19     if (currentScroll == maxScroll) bloc.add(PostEvent());
20   }
21
22   @override
23   Widget build(BuildContext context) {
24     return Scaffold(
25       appBar: AppBar(
26         title: Text('Infinite List Demo'),
27       ),
28     );
29   }
30 }
```

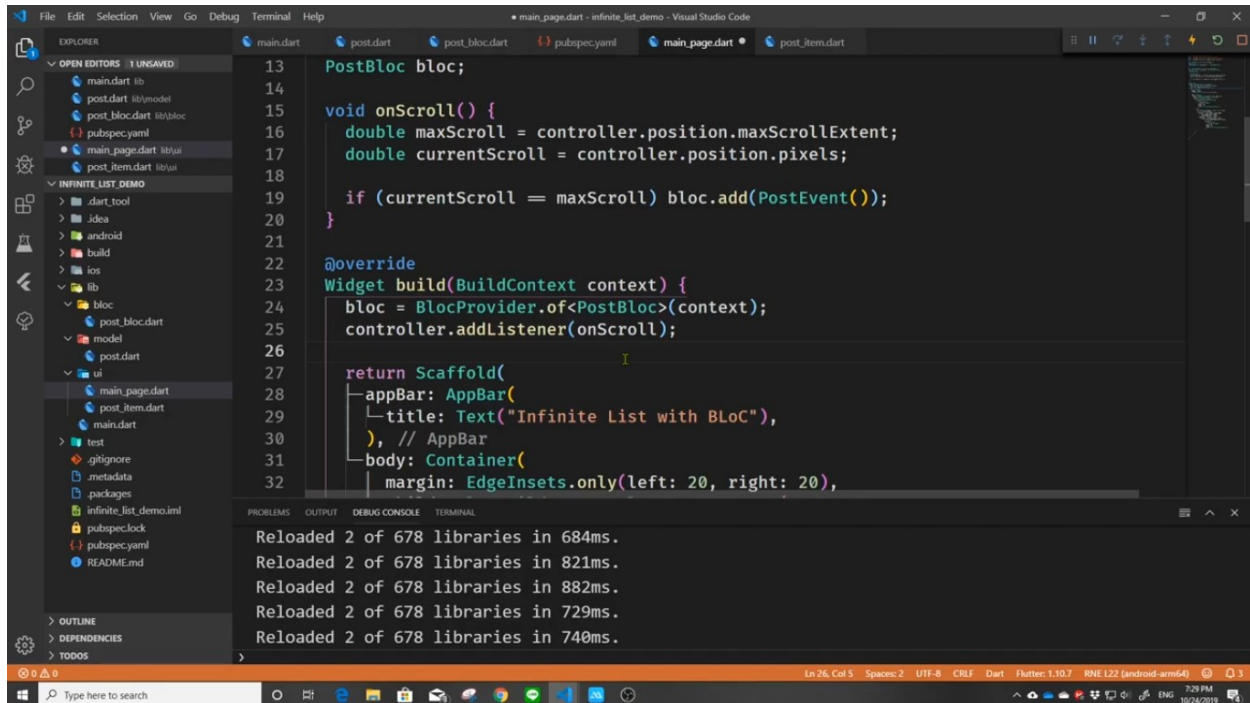
Reloaded 2 of 678 libraries in 684ms.
Reloaded 2 of 678 libraries in 821ms.
Reloaded 2 of 678 libraries in 882ms.
Reloaded 2 of 678 libraries in 729ms.
Reloaded 2 of 678 libraries in 740ms.

Lalu kita kebawh

Di context

Bloc = BlocProvider.of<PostBloc>(context);

Lalu `controller.addListener()` isi itu `onScroll`

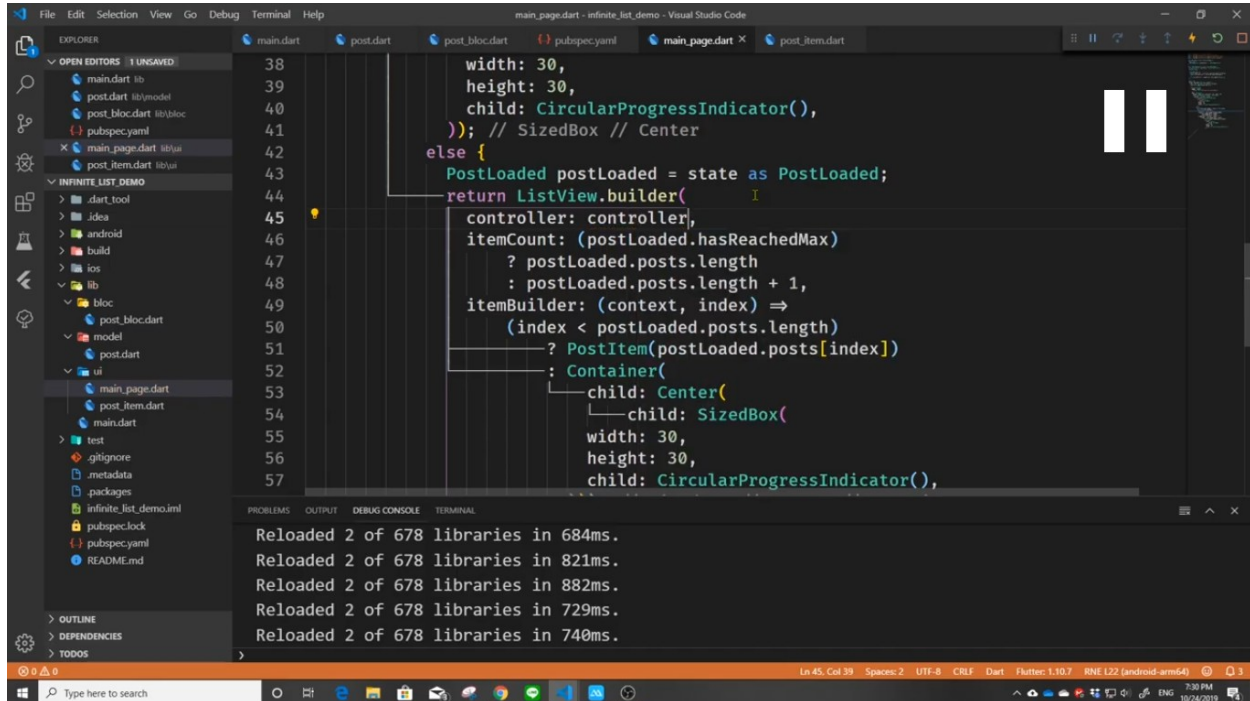


```
13 PostBloc bloc;
14
15 void onScroll() {
16   double maxScroll = controller.position.maxScrollExtent;
17   double currentScroll = controller.position.pixels;
18
19   if (currentScroll == maxScroll) bloc.add(PostEvent());
20 }
21
22 @override
23 Widget build(BuildContext context) {
24   bloc = BlocProvider.of<PostBloc>(context);
25   controller.addListener(onScroll);
26
27   return Scaffold(
28     appBar: AppBar(
29       title: Text("Infinite List with BLoC"),
30     ), // AppBar
31     body: Container(
32       margin: EdgeInsets.only(left: 20, right: 20),
```

Reloaded 2 of 678 libraries in 684ms.
Reloaded 2 of 678 libraries in 821ms.
Reloaded 2 of 678 libraries in 882ms.
Reloaded 2 of 678 libraries in 729ms.
Reloaded 2 of 678 libraries in 740ms.

Setelah itu kita daftar kan controller

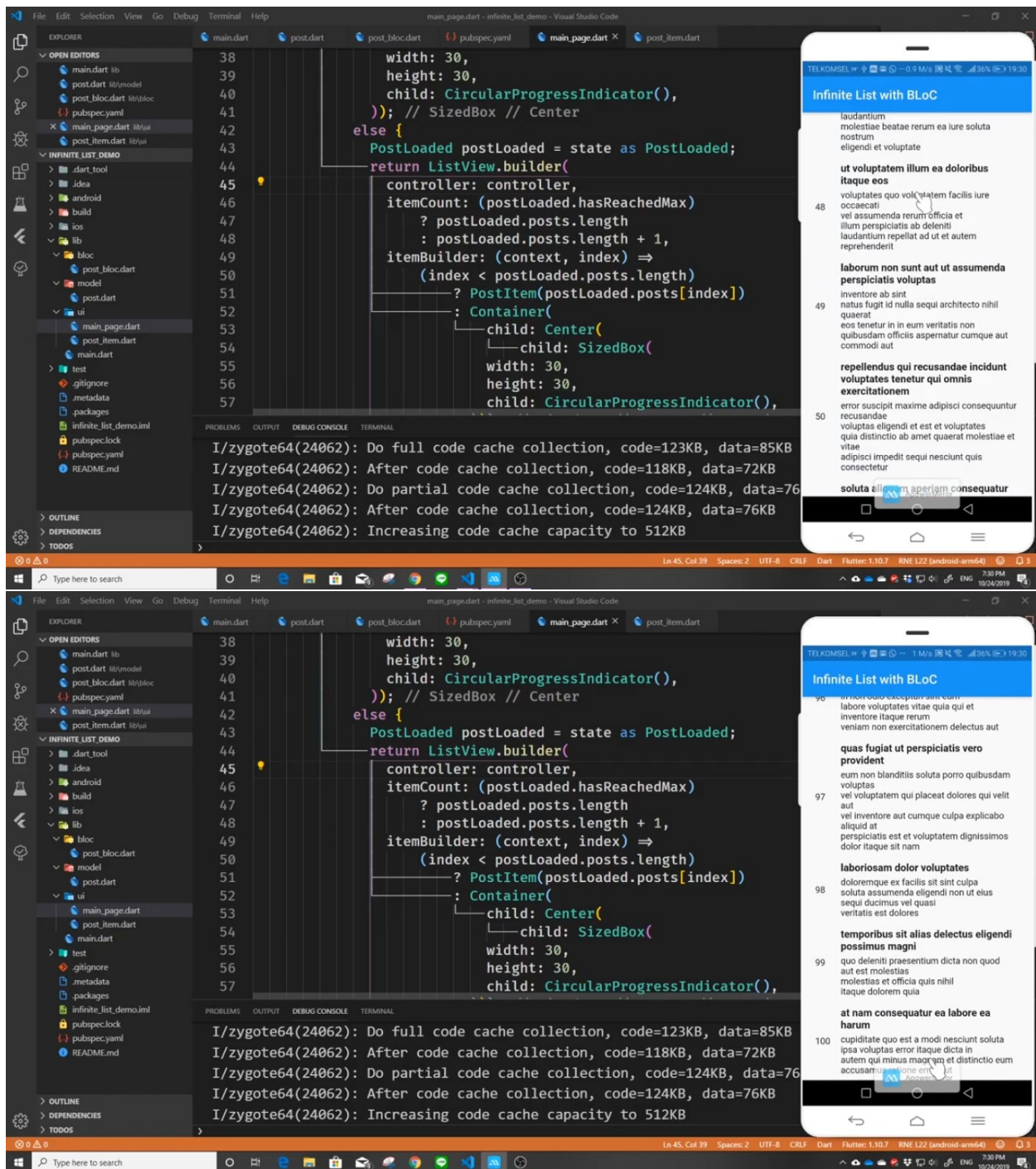
Controller = controller;



```
38 width: 30,
39 height: 30,
40 child: CircularProgressIndicator(),
41 ); // SizedBox // Center
42 else {
43   PostLoaded postLoaded = state as PostLoaded;
44   return ListView.builder(
45     controller: controller,
46     itemCount: (postLoaded.hasReachedMax)
47       ? postLoaded.posts.length
48       : postLoaded.posts.length + 1,
49     itemBuilder: (context, index) =>
50       (index < postLoaded.posts.length)
51         ? PostItem(postLoaded.posts[index])
52         : Container(
53           child: Center(
54             child: SizedBox(
55               width: 30,
56               height: 30,
57               child: CircularProgressIndicator(),
```

Reloaded 2 of 678 libraries in 684ms.
Reloaded 2 of 678 libraries in 821ms.
Reloaded 2 of 678 libraries in 882ms.
Reloaded 2 of 678 libraries in 729ms.
Reloaded 2 of 678 libraries in 740ms.

Maka kita lihat



Maka stop le 100 kita habis

Begini cara gunkan infinity dgn management state bloc